

Laravel Assessment Question - 100 Marks

Duration: 4 hours

I. Create Database Tables 5x2 = 10 Marks

1. Create a `employee` table with the following fields:
`id(uuid), name(string), email(string), department_id(int),
created_at(timestamps), updated_at(timestamps),
deleted_at(timestamps)`
2. Create a `employee_detail` table with the following fields:
`id(int), employee_id(uuid), designation(string), salary(float),
address(string), joined_date(date), created_at(timestamps),
updated_at(timestamps)`
3. Create a `department` table with the following fields:
`id(int), name(string), description(string),
created_at(timestamps), updated_at(timestamps),
deleted_at(timestamps)`
4. Use Laravel's migration system to create and run these tables.
5. Ensure your code is clean, well-structured, and follows Laravel conventions.

II. Insert Bulk Data 2x5 = 10 Marks

1. Insert 100,000 fake employees and 10 departments into the database.
2. Ensure your code is clean, well-structured, and follows Laravel conventions for large-scale data insertion.

III. Create Employee Management APIs 1x30 = 30 Marks

Develop full CRUD (Create, Read, Update, Delete) APIs for employee management. Implement security best practices. You will get an advantage if you write documentation/notation on APIs so that anyone could generate swagger easily in future.

IV. Admin Dashboard (Using Vue.js) 1x30 = 30 Marks

Build a Vue.js admin dashboard with full CRUD functionality for employee management. Include search, filter by salary range and department, and sort by joining date. Implement security best practices. Display Name, Email, Department, Designation, Joining date fields and edit & delete icons in the listing table view with pagination.

V. Application Performance Challenge 1x20 = 20 Marks

Ensure the application architecture is designed for high performance and can handle a large number of concurrent requests efficiently. Apply the necessary optimizations to meet performance expectations.